MULTISTEP ELECTRICITY PRICE FORECASTING FOR DEREGULATED ENERGY MARKETS: GAN-BASED DEEP REINFORCEMENT LEARNING

An Undergraduate Research Scholars Thesis

by

ALI EL ASSADI

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisors: Haitham Abu-Rub
Mohamed Massaoudi

May 2021

Major: Electrical and Computer Engineering

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Ali El Assadi, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Research Faculty Advisors prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

TABLE OF CONTENTS

ABSTRACT

Multistep Electricity Price Forecasting for Deregulated Energy Markets: GAN-Based Reinforcement Learning

Ali El Assadi
Department of Electrical and Computer Engineering
Texas A&M University

Research Faculty Advisor: Haitham Abu-Rub
Department of Electrical and Computer Engineering
Texas A&M University

Research Faculty Advisor: Mohamed Massaoudi
Department of Electrical and Computer Engineering
Texas A&M University

Electricity Price Forecasting (EPF) plays a vital role in smart grid applications for deregulated electricity markets. Most of the studies tend to investigate the electricity market influencers using forecasting techniques, often losing sight of significance on the sensibility of EPF models to the unstable real-time environment. This project will address a novel EPF based on deep reinforcement learning. The proposed approach uses generative adversarial networks (GAN) to collect synthetic data and increase training set effectively and increase the adaptation of the forecasting system to the environment. The data collected will be fed to a Deep Q learning to generate the final predictions. The proposed GAN-DQL will also be assessed on real data to prove the proposed model advantages compared to several machine learning solutions.

ACKNOWLEDGEMENTS

Contributors

I would like to thank my faculty advisor, Dr. Haitham Abu-Rub, and Eng. Mohamed Massaoudi for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

The New England Deregulated Energy Prices used for Electricity Price Forecasting were provided by Eng. Mohamed Massaoudi. The analyses were conducted in part by Electrical and Computer Engineering Department and these data are unpublished.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Note: I have not received funding for this research

1. INTRODUCTION

Over the last decades, the antiquated electric grid dominates the electrical market from production and transmission to distribution. However, the ongoing increase of the electricity demand and the vast adoption of new technologies in the vein of energy production such as renewable energy implementation boosts the energy producers to upgrade their energy market strategies towards a competitive market called also Deregulated Electricity Market (DEM) [1]. DEM is one of the essential elements of smart grids. The stockholders and energy producers participate to energy markets with their competitive selling bids. The consumers select the minimum selling prices from electric power industries [2], [3]. The markets operators choose the best buying prices according to their financial profitability.

Both energy operators and consumers use Electricity Price Forecasting (EPF) in restructured power systems to manage their bidding strategies [4]. EPF has gained significant attention in the research community due to its importance in power markets. Accurate EPF is mandatory to derive the best bidding strategies and maximize their profits for producers and utilities. However, designing a perfectly tailored forecasting technique to EPF presents a hard task with the nonlinear and stochastic electrical price. The variability of electrical prices is due to various parameters [5], [6]. These parameters include market' design, transmission congestion/ contingency, fuel price/ cost of unit operation, participants' bidding strategies, balance of supply and demand, weather conditions, seasons of the year, outages of large power plants, etc. [7]

Various algorithms are proposed to tackle the EPF. The energy hub focuses on contributing to the body of knowledge with the most accurate technique with minimum required computational burden. Machine learning techniques are deeply involved such as Long Short-Term Memory, Convolutional Neural Network, Deep belief network [8], [9]. However, most of the proposed machine learning techniques requires a large data for training. This data needs a high computational effort which make it very difficult to be implemented on online. Recent studies tend to use Reinforcement Learning (RL) techniques due to its inherent potential to deal with unseen scenarios [10], [11], [12]. However, very few researches try to implement RF for EPF. The main motivation behind this project is the need to have an accurate Electricity Price Forecasting (EPF) for online implementation in deregulated electricity environment.

The project will be beneficial for electrical industries since it will compensate for the lack of large real data from the energy market by generating artificial data that contains the same characteristics and follows the same trends. The proposed project will also precisely identify the prediction accuracy of the forecasting engine in online data preprocessing. The proposed project will also improve the profitability of the stakeholders in his energy market using an accurate prediction tool. The EPF model is lies in the hybridation of two powerful types of models: Generative Adversarial Network (GAN) and Reinforcement Learning

2. METHODS

2.1 Benchmarking

Currently, EPFs are developed using numerous machine learning techniques. These machine learning techniques are often limited by several factors that lead to inaccurate results. These inaccurate EPF models are not sufficient enough for smart grid applications as the supply companies need to accurately assess the needs of the consumer thus providing him with the best solution financially.

This project proposes a new electricity price forecasting model based on integrating two machine learning technologies: Reinforcement Learning and GAN. The proposed project will also precisely identify the prediction accuracy of the forecasting engine in online data preprocessing. The proposed project will also improve the profitability of the stakeholders in his energy market using an accurate prediction tool.

2.1.1. Benchmarking Criteria

In order to compare and benchmark the different existing solutions as well as our own solution, benchmarking criteria was set up. The main criteria that was used to evaluate the different solutions are listed below.

2.1.1.1. Machine Learning Technique Used

For this criteria, different types of machine learning techniques were studied. Furthermore, several existing models which implemented these different machine learning techniques were researched in order to find out the benefits and limitations of each machine learning technique. Based on the research, existing models which combine specifically GAN and deep reinforcement learning to generate EPFs could not be found. It is believed that these two techniques can be perfectly combined with each other as they complement each other very well.

2.1.1.2. Mean Absolute Error

The mean absolute error of the predictions to the actual value. The formula used to calculate this mean absolute error is shown below.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{1}$$

MAE = mean absolute error

y_i = prediction

x_i = true value

n = total number of data points

2.1.1.3. Limitations

We looked at several limitations of different types of existing models. Some of these limitations are limitations with the machine learning technique while other limitations are more general such as physical and economical limitations.

2.1.1.4. Economic factors

This is very important criteria as the cost of implementation must be evaluated as the entire point of electricity price forecast is to save money. If the cost to implement an EPF generating model is more than the money saved by having accurate EPFs then the solution is not economically feasible. Generating accurate EPFs for a low price will have a positive overall effect on the economy as although there might be a cost to implement these models/generate EPFs, the money saved will outweigh the initial cost in the long run. Furthermore, with the existence of accurate forecasts, governments can know when is the correct time to sell electricity at the peak price instead of losing out on potential revenue.

2.1.1.5. Global Impact

We looked at the solutions and evaluated their global impact and how feasible they are to implement worldwide. We found that since our proposed model is linked to a smart grid which is the future of electric grids, it is preferable to the current existing models. Smart grid is an electric grid that bases its whole functionality on several factors with the aim of using renewable energy resources and reducing wasted energy to have a positive impact on the environment.

2.1.1.6. Social Impact

In order to proceed with our solution, we need to check if it will be accepted by society and the social impact it would have. Our solution is closely linked to a smart grid which is an electric grid that heavily relies on a two-way communication between the consumer and the supplier for proper functionality. This factor ensures a clear communication between the customer and the distributor/producer. Yet, this communication helps in saving an enormous amount of energy with the satisfaction of both sides.

2.1.2. Benchmarking Table

Table 2.1 General Comparison and Performance

	Existing Solution	**Existing Solution**	**Existing Solution**	**Existing Solution**	**Existing Solution**	**Our Solution**
Technique Used	Artificial Neural Network (ANN) and Extreme Learning Machine (ELM) compared with simple moving average (SVA) **[1]**	Extreme Learning Machine-Bootstrap Electricity in market clearing prices framework **[2]**	Bayesian clustering by dynamics (BCD) and support vector machine (SVM) **[3]**	Deep recurrent neural networks (DRNN) **[4]**	Support Vector Machine (SVM), Artificial Neural Networks (ANN), Ant Colony Optimization (ACO) and ACO-ANN **[5]**	Generative adversarial network (GAN) and Deep Reinforcement learning (D-RL)
Advantages	Yields a higher forecasting accuracy due to ANN can consider multiple input variables **[1]**	Efficiently quantify the level of uncertainty associated with point forecasts, which comply with the needs for market operation and risk management purposes **[2]**	High degree of effectiveness and efficiency in learning and prediction compared to other methods **[3]**	Vital potential of the deep learning approach for future development **[4]**	results showed that the combination of ANN with ACO outperformed the ANN without the ACO **[5]**	High precision in forecasting Deregulated Energy market prices
Disadvantages and limitations	Hardware dependence and unexpected behavior of the network**[29]**	Uses several hidden layers that could affect the generated data, resulting in fault data	SVM is not suitable for large data	It is not always easy to utilize DRNN, because a continuum of data are required to set the initial condition **[30]**	SVM could underperform when the number of features for each data point exceeds the number of training data samples	-

Mean Absolute Percentage Error	1.025% for ELM and 1.035% for SMA	1.5374% for Short-Term time series 3.1276% for Long-Term time series	Average = 6.08%	Average = 5.63%	12.43% for SVM 9.4% for ACO-ANN 8.4% for ANN	-

2.1.3. Benchmarking Study Analysis and Summary

There are several existing solutions collected from scientific papers that have been evaluated using specific criteria. Moreover, the project evaluated using the same evaluation criteria and compared with the existing solutions for its performance and accuracy in predicting the electricity price. From the comparison tables above, it is clear that Bayesian clustering by dynamics (bcd) [23] and support vector (svm) [22] is the most similar model to the project. Where the svm is an algorithm that takes advantage of both regression or categorizing obstacles. The bcd aim is to have identical dynamics by categorizing the time sets into clusters. The advantage of using this technique is its high efficiency in learning and predicting, which makes it more suitable for epf. There are drawbacks of using this method, where the svm can have poor performance when the number of transing sets overtake the number of characteristics for each data. Therefore, it is not applicable for large numbers of data. The model has a mean value of 6.08%.

From the comparison table (Table 2.1), using deep recurrent neural networks (DRNN) to predict the electricity price forecasting for the day ahead can be another comparable solution. The deep recurrent neural network is an altered model of the entirely connected DRNN with self-recurrent neurons layer. There are two types of DRNN that are applied to a control system, the first one is an identifier and the second is a controller. The identifier is a diagonal recurrent neural identifier (DRNI) and the controller is a diagonal recurrent neural controller (DRNC) **[24]**. Moreover, the system dynamic attitude can be apprehended using DRNN. Therefore, DRNN has a fast learning rate because it involves both DRNI and DRNC. From the benchmarking tables, it is shown in Table 2.1 that the DRNN has a crucial potential of deep learning approach for future development. The doubt of using DRNN, that it requires a continuum of data to set initial conditions. Therefore, applying DRNN is crucial. The model has a mean value where taken for both months March and August which equals 5.63%.

The ANN, ELM, SVA, and hybrid networks are also integrated in these existing solutions, these machine learning algorithms have a good efficiency and accuracy. These solutions have disadvantages that decrease their performance as hardware dependency, inapplicable attitude of the network, and error data can be detected because of using several hidden layers.

Comparing it with our solution, it is shown that integrating SVM and BCD is the most similar method to ours. Where this project uses GAN and reinforcement learning algorithms. In our project, the GAN is used for generating new data based on available data and the reinforcement produces the final prediction by taking the new data generated by GAN.

2.2 Functional Modeling

In this assignment, a functional structure of our project design will be developed. Due to the complexity of the project, using graphical representation techniques such as flow-charts and black boxes will aid us greatly as it allows us to visualize the different steps of our project. Furthermore, these graphical representations make it very easy to identify possible difficulties we might face in our project model and how to overcome them so that we do not run into them during the implementation stage.

2.1.2. *Upper Level Functional Model:*

Figure 2.1 Black Box

In the black box Figure 2.1, the system in the design project is a combination of Generative adversarial network and deep Q-learning. These are two types of reinforcement learning; they can increase the accuracy of the electricity price forecasting (EPF). Moreover, GAN collects data, increases the training set and the adaptability of the system. The data collected from GAN is used in Deep Q-learning to generate final predictions for the EPF.

2.1.3. *Detailed Functional Modeling*

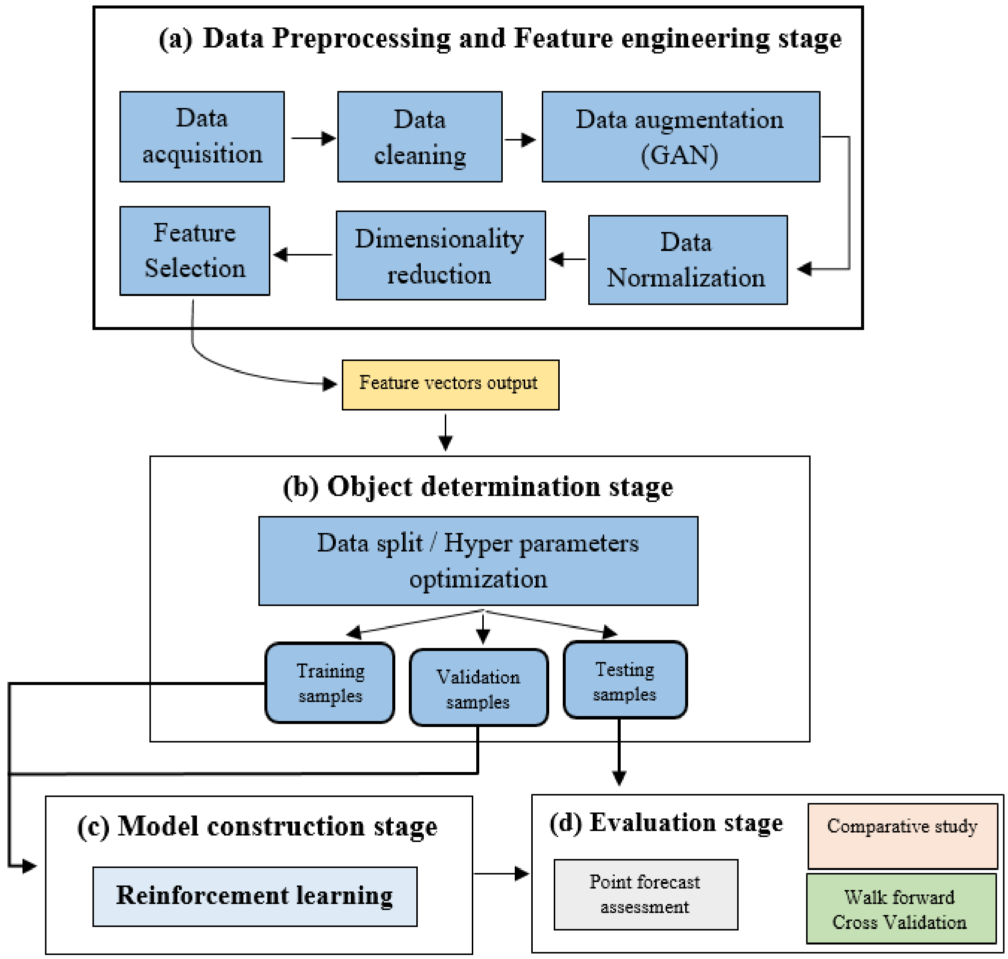

Figure 2.2 Architecture of the Project

For the project design detailed functional model, the project will be broken down into four major stages. The stages are explained below:

2.2.2.1. Stage a: Data Preprocessing and Feature engineering

In the first stage, the information is processed with data cleaning, feature extraction, and feature selection. The real data is extracted, acquired, and goes through a series of data cleaning in order to have the input environment ready.

- Data acquisition: The collected data was based on data from energy prices from the New England Electricity market.
- Data cleaning: Data cleaning or cleansing is the process of detecting and adjusting fault and inaccurate data from the acquired data.
- Data augmentation: It is used in order to increase diversity of the acquired data without searching for extra data. Data augmentation is often used in training large applications of neural networks. Generative adversarial network operates in this sub-function where it consists of two networks: Generative network and discriminative network.
- Data normalization: In order to enhance the integrity of the data, the data goes through data normalization which serves to set similar scales and ranges between the given feature data. Usually, it is not necessary in machine learning models or assignments **[28]**.
- Dimensionality reduction: Is a technique used in reducing the data from a high-dimensional to low-dimensional, reducing the number of input variables in the dataset.
- Feature selection: Also known as variable selection, is a technique used mostly in machine learning models in order to select specific and relevant features for a smart learning **[25]**.

2.2.2.2. Stage b: Object Determination

The dataset, more specifically: temperature, month, day, hour and load feature vectors are integrated into the object determination stage. This stage contains the problem formulation, where the data is split into training and testing.

- Data Split/Hyper Parameters Optimization: This sub-function serves in machine learning by choosing optimal data in order to control the learning process.
- Training/Validation/Testing Samples: Training samples are the actual samples used for the agent learning, validation samples are samples used to generate impartial and objective

judgement on the training set of the model, and testing samples are also generating impartial judgement of a training set of the final model.

2.2.2.3. Stage c: Model Construction

In this stage the trained and tested data is input to the Reinforcement learning technique in order to validate the output of the generative adversarial network and generate the final prediction. The reinforcement learning technique is called Deep Q-learning. Deep Q-learning is a type of machine learning that is based on an agent's states and actions. The agent starts by choosing arbitrary actions concerning its environment where it gets rewards or penalties specified to its actions. The programmer sets policies for the computer as rules to follow where these policies comprise rewards and penalties for artificial intelligence, depending on the actions performed.

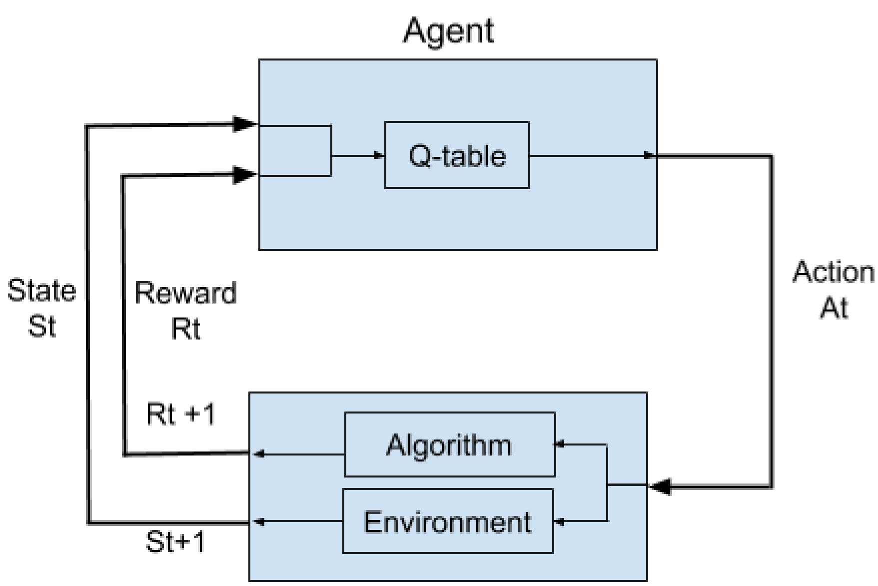

Figure 2.3 RL Construction

2.2.2.4. Stage d: Evaluation

The evaluation stage has three sub-functions: point forecast assessment, comparative study, and walk forward cross validation.

- Point forecast assessment: It is an assessment of point forecasting where methods are compared by means of an error measure or scoring function **[26]**.
- Comparative study: The aim of the comparative study is to identify similarities and differences between the generated and input data **[27]**.
- Walk forward cross validation: Cross validation is a technique used for taking two segments from the data, one for optimizing the system of the model and the other for validating.

2.3 Detailed System Design

Q-Learning is an RL Branch that deals with the same learning type in stimulating a specific environment through actions in different states. Q-learning dedicates its learning using a Mathematical Algorithm in which transmission from one state to another through a sequence of actions initiates a Q-value stored in a lookup table called: Q-table. Q-table is a table formed from the number of states and actions available in the environment (the rows and columns are often states and actions, respectively). The agent uses the Q-table for finding the maximum expected future rewards for actions in each state. Q-learning is used commonly in a game-like type of environment; on the other hand, it is also used in some applications in price forecasting and stock predictions.

Table 2.2 Q-table

		ACTIONS			
	Q(s,a)	LEFT	RIGHT	UP	DOWN
STATES	0	0.9	0.5	-0.4	-0.3
	1	-0.1	0.115	0.322	0.5
	2	-0.2	0.225	-0.66	0.8
	3	-0.8	0.77	-1.12	1.67

Table 2.2 is an example of a Q-table for a particular game where the agent has 4 states and can move in four different directions (left, right, up, and down). At the beginning of training the agent, the Q-table will be vacant from the Q-values. The values presented in Figure 1 are the Q-values at the end of the training, and as explained before, the agent uses this Q-table as a reference to play the game having the ultimate knowledge of playing it. For instance, if the agent in Table 2.2 was in state 2, it will know that the best action to take is to go DOWN, that is because moving DOWN from state 2 has given the agent a higher reward than the other three actions.

2.3.1. Algorithm

Q-learning uses a function to evaluate values in the Q-table. This function serves to calculate the Q-value, which is the input for the Q- table. The Algorithm works by using an old Q-value, learning rate, discount factor, and Q-estimate to calculate the New Q-value. The program stores and updates these Q-values into the Q-table.

In this application, RL is used with Generative Adversarial Network to build Short a Term Electricity price forecasting model in a deregulated Electricity Market. RL is fed with data collected from the GAN-Based model and marks the final predictions compared to the real output.

$$Q_{new}(s_t, a_t) = Q(s_t, a_t) + \alpha * \{ r_t + \gamma * \max Q(s_{t+1}, a) - Q(s_t, a_t) \} \quad (2)$$

Q is the Q-value

S_t and a_t are the state and action at a specific time respectively

α is the learning rate

r_t is the reward at a specific time

γ is the discount factor

The proposed project will be broken down into seven major stages/steps. The first step of the project is to collect as much data as possible about the electricity market. The next step is to use this data as a dataset and generate new data of the market using a GAN based solution. Next, the reinforcement learning environment, states, actions, and rewards are going to be identified according to the EPF problem. After that, we are going to implement the system in a programming platform (Python), evaluate its performance, and compare it to other benchmarks.

Finally, a Raspberry Pi computer is going to be used to implement the proposed model which will be tested with real data processing and reported in a scientific document.

In the proposed project, there already exists data sets of deregulated energy prices from 2011 till 2018 along with their prices.

In our solution, this data is going to be used as an input for the GAN, where the discriminator will train the given data while the generative network produces artificial data similar to this input data. The discriminator will then estimate the accuracy of the artificial data while the generator continues training from the output of the discriminator. This creates a loop between the two networks where one learns from the other.

After implementing the first machine learning algorithm, the output data from the Generative Adversarial Network is fed to the Reinforcement learning technique (Q-learning), that serves to produce the final prediction.

After feeding the data to Q-learning, an environment is created to facilitate the demands of the Q-learning algorithm (States, Actions). The deregulated energy values in this data serves as an environment for this Machine learning technique. The environment is the reference for the agent in which the execution of the learning takes place. The last column in the data is the actual prices of the deregulated energy values consumed (DA_LMP). The number of states is the count of the different values of the actual cost, and the actions are an array with negative and positive values depending on the minimum and maximum cost values. After finding the average number (AVG), the minimum (Min), and the maximum number (Max) of the energy cost column, the actions array is created with successive integer values starting from {Min - AVG} ending with {Max - AVG}. Whereas, the average number of the cost values will serve as the initial state for the agent. For example, if the calculated average value is 8 units, and the minimum and the maximum cost values are 3 units and 12 units respectively. Thus, the actions array will be generated as follows: {-5, -4, -3, -2, -1, 0, 1, 2, 3, 4}, and the initial state of the agent is 8. The whole purpose behind this reach is to pave the way for the agent to take a series of actions and get rewarded or penalized on his actions. After starting in the initial state, the agent takes a series of random actions that are valued by the Q-learning algorithm and stored in the Q-table.

2.4 Analysis of the program code

The program code was implemented on google colab for the software has the libraries needed and it gives a better editing access to all the group members. The code started by importing the needed packages like pycaret, Keras, scikit-learn, sklearn, and Pytorch. After that, from these packages, we imported almost every library that we could use in order just to avoid programming errors.

Moving on with the program code, we imported the database for our project. The experimental settings mainly use the ISO New England incorporation (ISO-NE) dataset. The ISO-NE is non-profit incorporation to manage day-to-day operations with more than 200 market participants in New England. The database can be downloaded on open sources in **[32]**.

After Importing the database, the csv file, and reading the csv file, we move on to data preprocessing. In Machine Learning models, data preprocessing is when the data is transformed, to ensure a better understanding for the Algorithm. In other words, it is used to let the model better read the input data.

```
[ ] from sklearn.linear_model                  import ElasticNetCV, ElasticNet

    infoo=xx.info()
    print(infoo)
    infoo_NAN1=xx.isnull().sum()
    print(infoo_NAN1)
    x_weathertrain_clean=xx.dropna()
    infoo1=x_weathertrain_clean.info()
    print(infoo)
    infoo_NAN=x_weathertrain_clean.isnull().sum()
    print(infoo_NAN)

    inp_train=pd.DataFrame(x_weathertrain_clean)

    data=xx.describe().T

    x=inp_train.drop(['DA_LMP'],axis=1)

    x=x.values

    y=inp_train.drop(['year', 'month', 'day', 'weekday', 'season', 'weeknumber', 'weekend', 'Hr_End', 'DA_Demand', 'RT_Demand', 'DA_EC', 'DA_CC', 'DA_MLC', 'RT_LMP', 'RT
    y=y.values
```

Figure 2.4 Data Preprocessing

Moving forward with the code, from sklearn.model_selection we have selected the number of training hours and we split the data into input and outputs. After implementing the train, value, and test arrays into x and y, we set the train ration, the validation ratio, and the test ratio to 0.75, 0.15, and 0.10 respectively. This illustrates that the training is 75% of the entire dataset, the test and the validation are 10% and 15% respectively of the initial dataset. We also reshaped the input X training, testing, and validation. The next step is model construction.

Furthermore, we began to train the model through model.fit, where we use Epoch for the iterations. Each Epoch takes around 4 to 6 seconds to train, and we choose the model to go through 100 Epochs. The training estimation time was 500 seconds which is around 9 minutes. It works by saving the validation loss from each iteration into the checkpoint, and using the new value to the next iteration. The model started at Epoch 0 with a loss of 389.7387 to 2.2459 at Epoch 100. *Figure 2.5* that shows the first 5 Epochs of the iteration.

```
Epoch 1/100
1169/1169 [==============================] - 6s 4ms/step - loss: 389.7387 - val_loss: 74.4432

Epoch 00001: val_loss improved from inf to 74.44316, saving model to 23_checkpoint.keras
Epoch 2/100
1169/1169 [==============================] - 4s 3ms/step - loss: 95.7005 - val_loss: 53.5919

Epoch 00002: val_loss improved from 74.44316 to 53.59193, saving model to 23_checkpoint.keras
Epoch 3/100
1169/1169 [==============================] - 4s 3ms/step - loss: 50.3361 - val_loss: 48.8154

Epoch 00003: val_loss improved from 53.59193 to 48.81539, saving model to 23_checkpoint.keras
Epoch 4/100
1169/1169 [==============================] - 4s 4ms/step - loss: 45.5865 - val_loss: 41.1650

Epoch 00004: val_loss improved from 48.81539 to 41.16497, saving model to 23_checkpoint.keras
Epoch 5/100
1169/1169 [==============================] - 4s 3ms/step - loss: 37.6230 - val_loss: 32.3249

Epoch 00005: val_loss improved from 41.16497 to 32.32494, saving model to 23_checkpoint.keras
Epoch 6/100
1169/1169 [==============================] - 4s 4ms/step - loss: 26.4822 - val_loss: 30.2422
```

Figure 2.5 First Five Epochs of the iteration

Functional Prototyping, Testing, Troubleshooting, and Experimental Results. We have implemented the model on Google Colab. The experimental results are shown in the figures below:

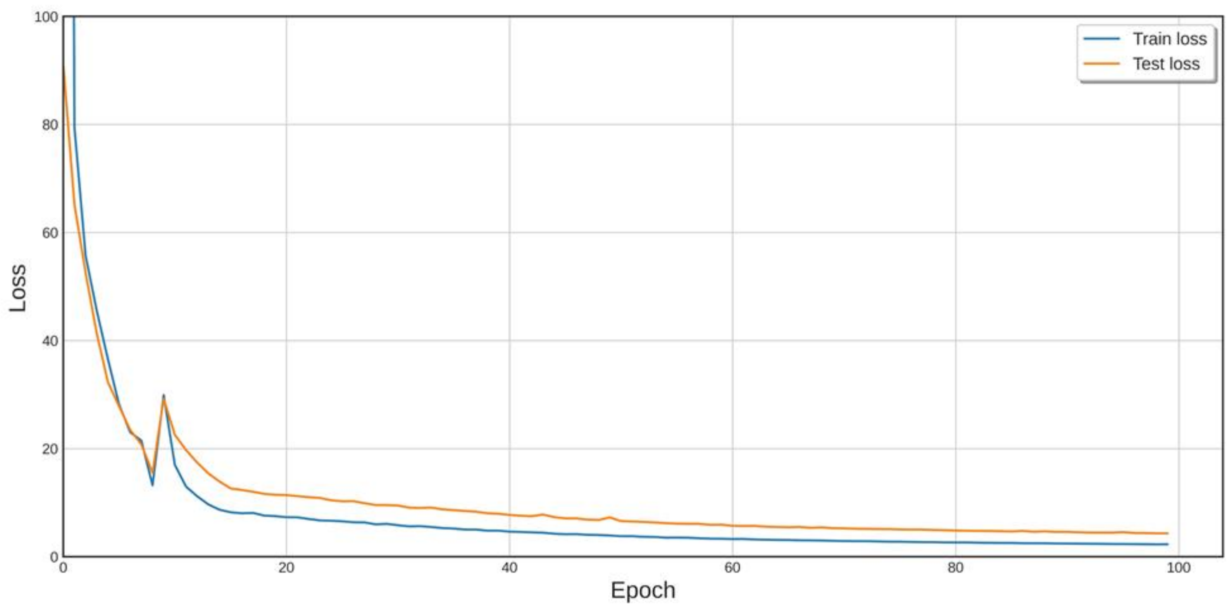

Figure 2.6 Epochs vs Loss graph

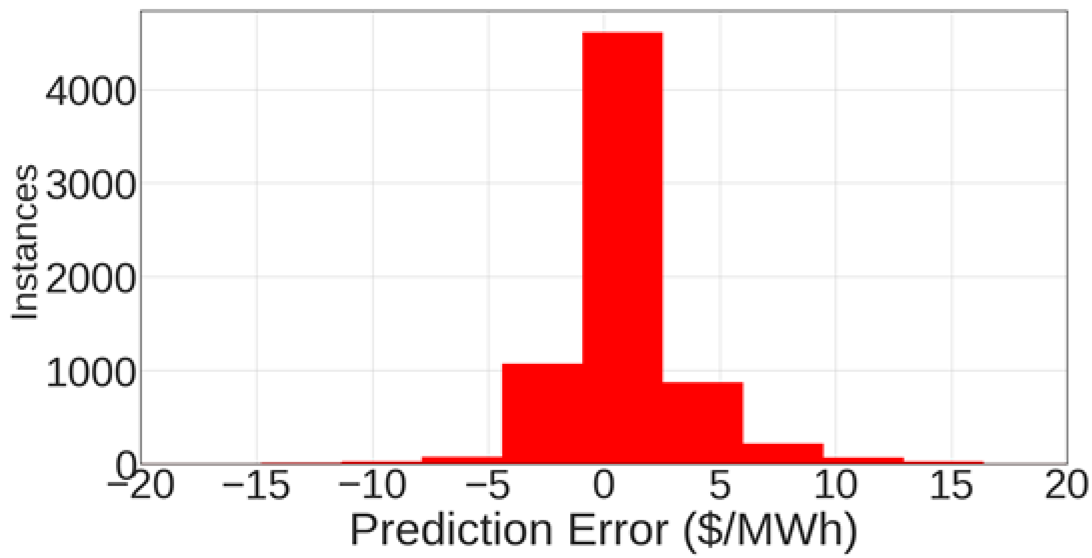

Figure 2.7 Prediction Error Experimental Result

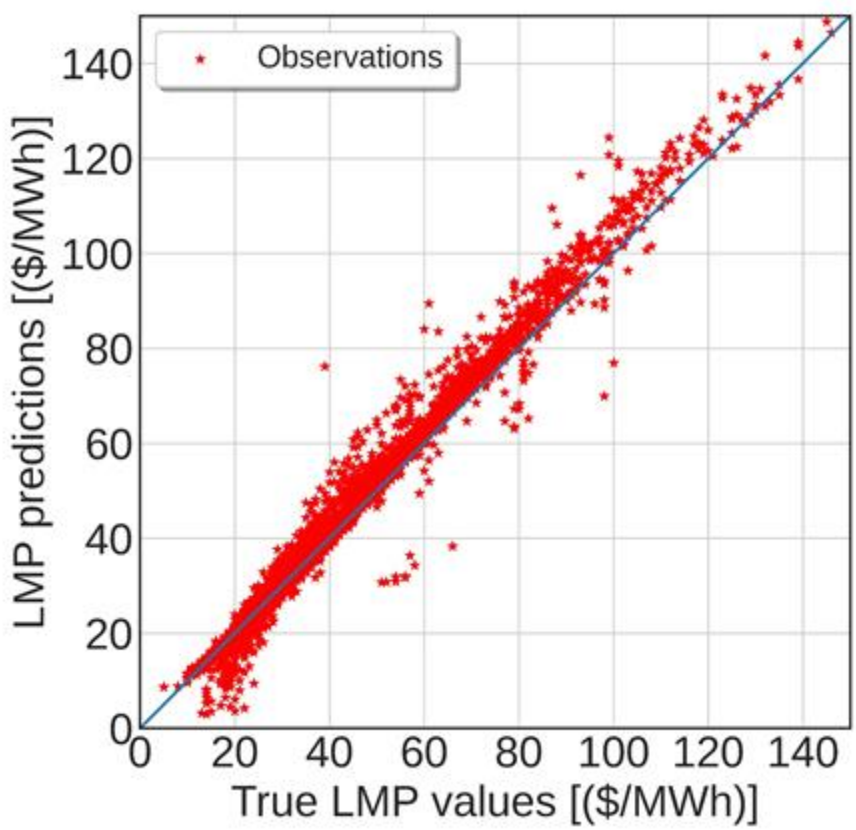

Figure 2.8 LMP Prediction Experimental Graph

Regarding the score evaluation, in regression tasks, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are commonly used metrics for ML models assessment. The RMSE represents the sample standard deviation of the differences between predicted values and observed values. The MAE is a linear score which means that all the individual differences are weighted equally in the average. MAPE is the mean absolute percentage deviation expressing accuracy as a percentage.

3. RESULTS

3.1. Analysis and Evaluation

The goal of the functional modeling is to implement the project design by identifying the input, system, and output. Therefore, to have a clear visualization of the project a black box and detailed functional model are introduced. The detailed functional model of the project shown in **Fig.2**, determined all the stages/steps needed for the project to achieve its main objective. The main objective of the proposed project is to integrate two machine learning methods to achieve an accurate electricity price forecasting (EPF). There are four main stages for the project, which are data processing and feature engineering, object determination, model construction, and lastly evaluation. These stages are identified carefully to have an accurate EPF which is our project.

At the start of this assignment, we set up a clear model of what input was needed in order to end up with the required output which was the EPF. The project can be broken down into four main stages. In the first stage, we have to acquire the data which will be cleaned and processed. Next, we will augment this data by using the Generative Adversarial Network. After that we will normalize the data produced by the Generative Adversarial Network and follow up with reducing the dimensions and performing the feature selection. In the following stage, we will split the data into three main groups: training samples, validation samples, and testing samples. In the third stage, we will construct the reinforcement learning model and feed the training and validation samples to the model. Finally, in the evaluation stage, we will evaluate our results using point forecast assessment, comparative study, and walk forward cross validation.

By completing this assignment and breaking down our project into these stages, we can know more about the specifics in each part of the project. For example, in stage a, when looking at it in general, it can be summarized as making sure the data is ready to be inputted into our model however by breaking it down into stages, we can see that it is more complicated than that. We first have to acquire/clean/augment/normalize the data. After that, we apply dimensional reduction to the data and finally, apply feature selecting. All of these stages were explained and we now have a better idea of exactly how to proceed in order to make the data ready for inputting. Similarly, in stage b, in general we are splitting the data however by completing this assignment, we know that we have to split them optimally using hyper parameters into three separate samples. In stage c, we broke down our model and showed how it will be constructed and how the Deep Q-learning technique will function and learn. Finally, we broke down the evaluation stage into its three sub functions: point forecast assessment, comparative study, and walk forward class validation. By completing this assignment and breaking down all these stages into separate tasks, we now know exactly what procedures to take and how to implement them in this project.

3.2. Technical Standards and Constraints

Using machine learning has several advantages that brings our project the efficiency needed to satisfy our results. Our group has studied the technical standards and constraints, where it has been found that there are some risks and constraints that need to be taken into consideration. On one hand, concerning the technical standards, constraints, and risks. The GANs go into details of data and can easily interpret into different versions so it is helpful in doing machine learning work. On the other hand, it has some common disadvantages, like for example it is hard to train where you need to provide different types of data continuously to check if it works accurately or not. The generating results from text or speech is very complex under GAN **[31]**.

In Deep reinforcement learning, we found out in our research that an RL model is likely to have an unexpected behavior once deployed. Also, the RL sometimes can lead to an overload of states, and that is due to the state-action algorithm of Deep Q-learning.

Working on tackling these constraints, our project uses Deep Q-learning which is one of the best machine learning techniques and that is due it visualizes the process within a table which is accessible by the user. This table is called Q-table, shown above in Table 2.2. Moreover, GAN has proven that such risks can be overcome by several programming approaches that are available online, hence ensuring that the project gives efficient results under different technical standards and constraints.

4. CONCLUSION

To conclude, the electricity statistical methods which are used by the electricity market influencers, conducted that sometimes it loses the sight of relevance on the vulnerability of the Electricity Price Forecasting models to the uncertain real-time domain. Therefore, Generative Adversarial Network (GAN) is a type of machine learning which is used to increase the training set, increase the adaptability of forecasting systems, and to collect data. The data that are used are collected from a real electricity market. Using deep Q learning, the data that have been collected will be generated as predictions. So, the new data is generated from the original dataset using GAN. By identifying the reinforcement learning environment, actions, states, and rewards according to the Electricity Price Forecasting dilemma the system will be programmed and evaluated for its performance.

REFERENCES

[1] J. E. C. Tee, T. T. Teo, T. Logenthiran, W. L. Woo and K. Abidi, "Day-ahead forecasting of wholesale electricity pricing using extreme learning machine," TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, 2017, pp. 2973-2977, doi: 10.1109/TENCON.2017.8228371

[2] X. Chen, Z. Y. Dong, K. Meng, Y. Xu, K. P. Wong and H. W. Ngan, "Electricity Price Forecasting With Extreme Learning Machine and Bootstrapping," in *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2055-2062, Nov. 2012, doi: 10.1109/TPWRS.2012.2190627.

[3] S. Fan, J. R. Liao, K. Kaneko and L. Chen, "An Integrated Machine Learning Model for Day-Ahead Electricity Price Forecasting," *2006 IEEE PES Power Systems Conference and Exposition*, Atlanta, GA, 2006, pp. 1643-1649, doi: 10.1109/PSCE.2006.296159.

[4] C. Zhang, R. Li, H. Shi and F. Li, "Deep learning for day-ahead electricity price forecasting," in IET Smart Grid, vol. 3, no. 4, pp. 462-469, 8 2020, doi: 10.1049/iet-stg.2019.0258.

[5] E. Foruzan, S. D. Scott and J. Lin, "A comparative study of different machine learning methods for electricity prices forecasting of an electricity market," *2015 North American Power Symposium (NAPS)*, Charlotte, NC, 2015, pp. 1-6, doi: 10.1109/NAPS.2015.7335095.

[6]"Energy Consumption," Energy Consumption - an overview | ScienceDirect Topics. [Online]. Available: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/energy-consumption. [Accessed: 22-Oct-2020].

[7] Sparkenergy J. 3, "Top Ten Most Electricity-Drawing Appliances and How to Save," Spark Energy, 14-Mar-2019. [Online]. Available: https://www.sparkenergy.com/appliance-electricity-use/. [Accessed: 22-Oct-2020].

[8] "ACs consume up to 70% of electricity in houses: Kahramaa," The Peninsula Qatar. [Online]. Available: https://www.thepeninsulaqatar.com/article/04/07/2017/ACs-consume-up-to-70-of-electricity-in-houses-Kahramaa. [Accessed: 22-Oct-2020].

[9] P. J. G. Pearson, "Environmental impacts of electricity generated by developing countries: issues, priorities and carbon dioxide emissions," in IEE Proceedings A - Science, Measurement and Technology, vol. 140, no. 1, pp. 100-108, Jan. 1993, doi: 10.1049/ip-a-3.1993.0017.

[10] H. Chitsaz, P. Zamani-Dehkordi, H. Zareipour and P. P. Parikh, "Electricity Price Forecasting for Operational Scheduling of Behind-the-Meter Storage Systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6612-6622, Nov. 2018, doi: 10.1109/TSG.2017.2717282.

[11] H. Jahangir et al., "A Novel Electricity Price Forecasting Approach Based on Dimension Reduction Strategy and Rough Artificial Neural Networks," IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2369-2381, April 2020, doi: 10.1109/TII.2019.2933009.

[12] K. Hubicka, G. Marcjasz and R. Weron, "A Note on Averaging Day-Ahead Electricity Price Forecasts Across Calibration Windows," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 1, pp. 321-323, Jan. 2019, doi: 10.1109/TSTE.2018.2869557.

[13] A. Pourdaryaei, H. Mokhlis, H. A. Illias, S. H. A. Kaboli, S. Ahmad and S. P. Ang, "Hybrid ANN and Artificial Cooperative Search Algorithm to Forecast Short-Term Electricity Price in De-Regulated Electricity Market," *IEEE Access*, vol. 7, pp. 125369-125386, 2019, doi: 10.1109/ACCESS.2019.2938842.

[14] E. İlseven and M. Göl, "Hydro-Optimization-Based Medium-Term Price Forecasting Considering Demand and Supply Uncertainty," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4074-4083, July 2018, doi: 10.1109/TPWRS.2017.2771618.

[15] K. Wang, C. Xu, Y. Zhang, S. Guo and A. Y. Zomaya, "Robust Big Data Analytics for Electricity Price Forecasting in the Smart Grid," *IEEE Transactions on Big Data*, vol. 5, no. 1, pp. 34-45, 1 March 2019, doi: 10.1109/TBDATA.2017.2723563.

[16] Hu, Linlin, et al. "A review of short-term electricity price forecasting techniques in deregulated electricity markets." *2009 44th International Universities Power Engineering Conference (UPEC)*. IEEE, 2009.

[17] Nowotarski, Jakub, and Rafał Weron. "Recent advances in electricity price forecasting: A review of probabilistic forecasting." *Renewable and Sustainable Energy Reviews* 81 (2018): 1548-1568.

[18] Deb, Chirag, et al. "A review on time series forecasting techniques for building energy consumption." *Renewable and Sustainable Energy Reviews* 74 (2017): 902-924.

[19] Lu, Renzhi, Seung Ho Hong, and Xiongfeng Zhang. "A dynamic pricing demand response algorithm for smart grid: reinforcement learning approach." *Applied Energy* 220 (2018): 220-230.

[20] C. Feng, M. Sun and J. Zhang, "Reinforced Deterministic and Probabilistic Load Forecasting via Q -Learning Dynamic Model Selection," in IEEE Transactions on Smart Grid, vol. 11, no. 2, pp. 1377-1386, March 2020, doi: 10.1109/TSG.2019.2937338.

[21] A. Chiş, J. Lundén and V. Koivunen, "Reinforcement Learning-Based Plug-in Electric Vehicle Charging With Forecasted Price," in IEEE Transactions on Vehicular Technology, vol. 66, no. 5, pp. 3674-3684, May 2017, doi: 10.1109/TVT.2016.2603536.

[22] U. code), "SVM | Support Vector Machine Algorithm in Machine Learning", Analytics Vidhya, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/. [Accessed: 02- Nov- 2020].

[23] Ramoni, Marco; Sebastiani, Paola; and Cohen, Paul, "Bayesian Clustering by Dynamics" (2001). Computer Science Department Faculty Publication Series. 158. Retrieved from https://scholarworks.umass.edu/cs_faculty_pubs/158

[24] Chao-Chee Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," in *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 144-156, Jan. 1995, doi: 10.1109/72.363441.

[25] DeepAI, “Feature Selection,” *DeepAI*, 17-May-2019. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/feature-selection. [Accessed: 09-Nov-2020].

[26] T. GNEITING, “Making and Evaluating Point Forecasts,” *JSTOR*, 2011. [Online]. Available: https://www.jstor.org/stable/41416407?seq=1. [Accessed: 2020].

[27] M. S. Lewis-Beck, “Comparative Research,” *Sage Research Methods*, 2004. [Online]. Available: https://methods.sagepub.com/reference/the-sage-encyclopedia-of-social-science-research-methods/n141.xml. [Accessed: 09-Nov-2020].

[28] T. A. I. Team, “How, When, and Why Should You Normalize / Standardize / Rescale Your Data?,” *Towards AI - The Best of Tech, Science, and Engineering*, 29-May-2020. [Online]. Available: https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff. [Accessed: 09-Nov-2020].

[29] M. Mijwil, "Artificial Neural Networks Advantages and Disadvantages", Linkedin.com, 2020. [Online]. Available: https://www.linkedin.com/pulse/artificial-neural-networks-advantages-disadvantages-maad-m-mijwel/. [Accessed: 02- Nov- 2020].

[30] I. Tokuda, R. Tokunaga, and K. Aihara, "Back-propagation learning of infinite dimensional dynamical systems," Neural Networks, vol. 16, pp. 1179-1193, 2003.

[31] J. Rehman, “Advantages and disadvantages of generative adversarial networks (GAN),” *IT Release*, 24-Jun-2020. [Online]. Available: https://www.itrelease.com/2020/06/advantages-and-disadvantages-of-generative-adversarial-networks-gan/. [Accessed: 06-Dec-2020].